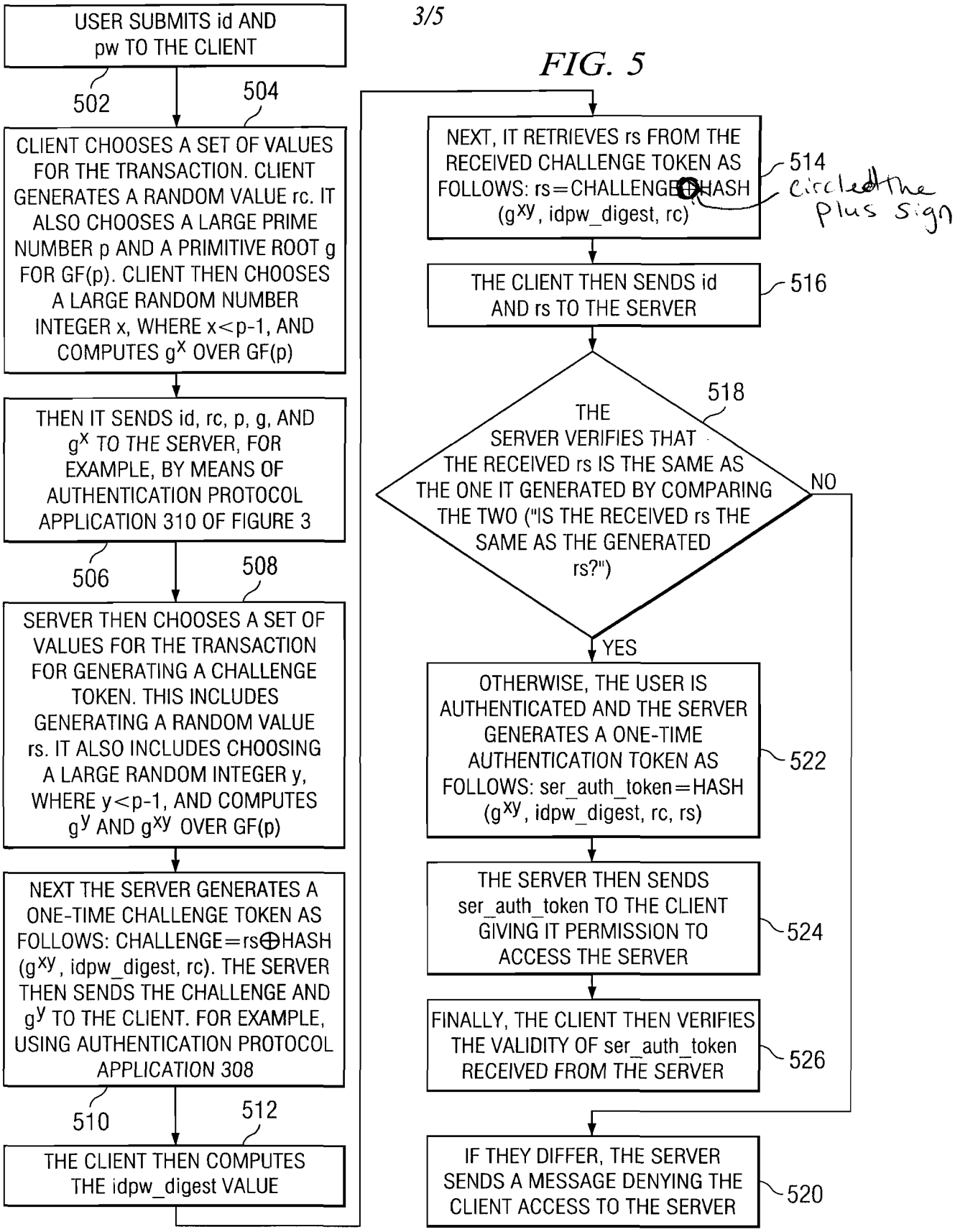
FIG. 5


USER SUBMITS id AND pw TO THE CLIENT
502
504
CLIENT CHOOSES A SET OF VALUES FOR THE TRANSACTION. CLIENT GENERATES A RANDOM VALUE rc. IT ALSO CHOOSES A LARGE PRIME NUMBER p AND A PRIMITIVE ROOT g FOR GF(p). CLIENT THEN CHOOSES A LARGE RANDOM NUMBER INTEGER x, WHERE x<p-1, AND COMPUTES g^x OVER GF(p)
THEN IT SENDS id, rc, p, g, AND g^x TO THE SERVER, FOR EXAMPLE, BY MEANS OF AUTHENTICATION PROTOCOL APPLICATION 310 OF FIGURE 3
506
508
SERVER THEN CHOOSES A SET OF VALUES FOR THE TRANSACTION FOR GENERATING A CHALLENGE TOKEN. THIS INCLUDES GENERATING A RANDOM VALUE rs. IT ALSO INCLUDES CHOOSING A LARGE RANDOM INTEGER y, WHERE y<p-1, AND COMPUTES g^y AND g^{xy} OVER GF(p)
NEXT THE SERVER GENERATES A ONE-TIME CHALLENGE TOKEN AS FOLLOWS: CHALLENGE=rs⊕HASH (g^{xy}, idpw_digest, rc). THE SERVER THEN SENDS THE CHALLENGE AND g^y TO THE CLIENT. FOR EXAMPLE, USING AUTHENTICATION PROTOCOL APPLICATION 308
510
512
THE CLIENT THEN COMPUTES THE idpw_digest VALUE
NEXT, IT RETRIEVES rs FROM THE RECEIVED CHALLENGE TOKEN AS FOLLOWS: rs=CHALLENGE⊕HASH (g^{xy}, idpw_digest, rc)
514
circled the plus sign
THE CLIENT THEN SENDS id AND rs TO THE SERVER
516
518
THE SERVER VERIFIES THAT THE RECEIVED rs IS THE SAME AS THE ONE IT GENERATED BY COMPARING THE TWO ("IS THE RECEIVED rs THE SAME AS THE GENERATED rs?")
NO
YES
OTHERWISE, THE USER IS AUTHENTICATED AND THE SERVER GENERATES A ONE-TIME AUTHENTICATION TOKEN AS FOLLOWS: ser_auth_token=HASH (g^{xy}, idpw_digest, rc, rs)
522
THE SERVER THEN SENDS ser_auth_token TO THE CLIENT GIVING IT PERMISSION TO ACCESS THE SERVER
524
FINALLY, THE CLIENT THEN VERIFIES THE VALIDITY OF ser_auth_token RECEIVED FROM THE SERVER
526
IF THEY DIFFER, THE SERVER SENDS A MESSAGE DENYING THE CLIENT ACCESS TO THE SERVER
520